

Tech Note

LHM

SonicWALL LHM Resource Center

Product

This document answers frequently asked questions about SonicWALL's implementation of Lightweight Hotspot Messaging (LHM) and describes the sample LHM scripts SonicWALL has created. This document contains the following sections:

- [FAQ](#)
- [LHM Script Library](#)

Note: The sample LHM scripts created by SonicWALL are available in this document:

http://www.sonicwall.com/support/pdfs/technotes/LHM_Scripts.pdf

The LHM syntax itself is defined in this document:

<http://www.sonicwall.com/support/pdfs/technotes/LHM.pdf>

FAQ

1. [What is LHM?](#)
2. [Do the LHM server scripts have to be written in ASP?](#)
3. [Why were these new scripts written in ASP.NET?](#)
4. [How can I use LHM to provide Guest Services access to wired users?](#)
5. [What is the difference between "authentication" and "authorization"?](#)
6. [Can I use LHM to provide access using \[LDAP, RADIUS, a button, the time of day, tasseography, a survey, relative barometric pressure, a passcode, etc.\] as the authenticator?](#)
7. [Can SonicWALL write the script for me that will do that?](#)
8. [I want to use the sample scripts SonicWALL provided. What do I need to do to use them?](#)
9. [Where can the LHM server reside?](#)
10. [Why are my guest clients unable to reach the LHM Server?](#)
11. [Why are the pages on the LHM server not loading?](#)
12. [How does the LHM exchange between the SonicWALL and the LHM server work? \(Concise version, typical environment\)](#)
13. [What do all the LHM settings mean? How do I configure them?](#)
14. [Can I change the LHM Management port from its default of TCP 4043?](#)
15. [Do I need to use the HMAC option? If I do want to use it, how do I use it?](#)
16. [Does SonicWALL provide any support for these scripts?](#)
17. [I've written a new script, I've made some great enhancements to your scripts, or I've just made your scripts work a whole lot better than you did – is SonicWALL interested?](#)



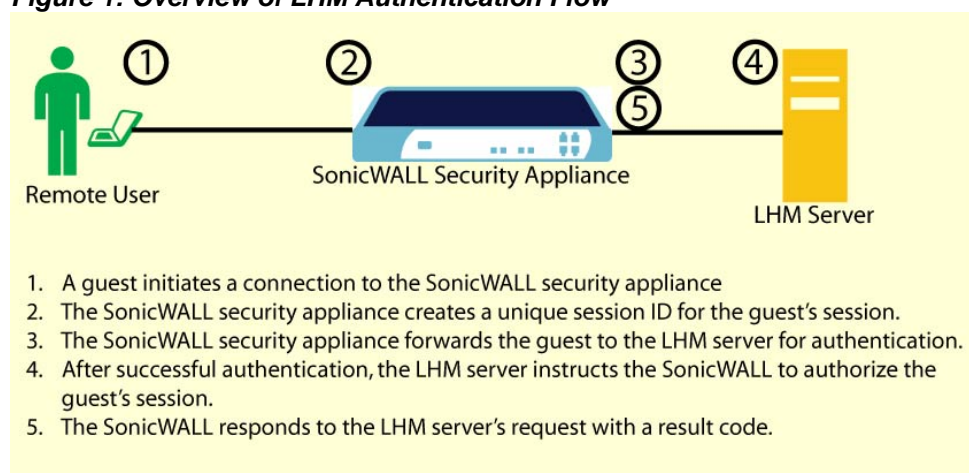
Tech Note

1. What is LHM?

LHM stands for Lightweight Hotspot Messaging. LHM leverages the SonicWALL Guest Service model, wherein users can be classified and authorized for differentiated network access through a SonicWALL security appliance. For example, the SonicWALL can be configured such that any user connecting through an interface belonging to a guest-services enabled WLAN (wireless LAN) Zone will only have access to the Internet (Untrusted network), but will not have access to the LAN (Trusted network). This allows a single network appliance to offer simultaneous access to trusted and guest users.

LHM extends the Guest Services model by breaking apart the authentication and authorization processes (described in FAQ entry #5), allowing the authentication to occur external to the SonicWALL. This allows for extensive customization of the authentication interface, and also allows for any kind of imaginable authentication scheme to be used.

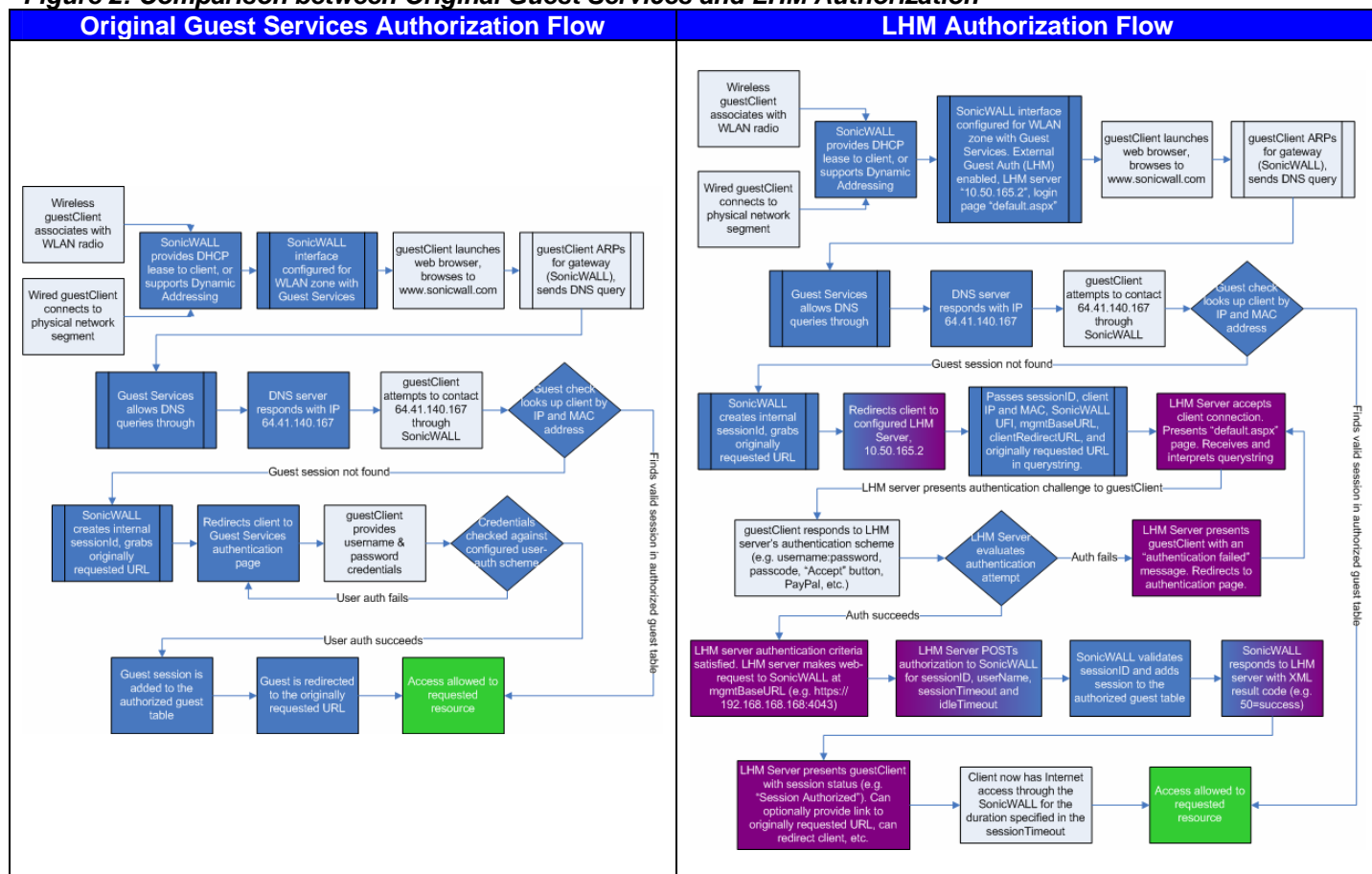
Figure 1: Overview of LHM Authentication Flow



Tech Note

Figure 2 shows how the process of LHM authorization differs from the original Guest Services authorization.

Figure 2: Comparison between Original Guest Services and LHM Authorization



The LHM syntax itself is defined in the document: <http://www.sonicwall.com/support/pdfs/technotes/LHM.pdf>

2. Do the LHM server scripts have to be written in ASP?

No – the LHM server scripts can be written using any platform capable of handling web requests and XML, the two core components of LHM. This includes Perl, PHP, ASP, ASP.NET, and J2EE.

3. Why were these new scripts written in ASP.NET?

ASP.NET was chosen for the new scripts because of its prevalence, and because it does lots of things well, not the least of which being the ease with which it handles XML.



Tech Note

4. How can I use LHM to provide Guest Services access to wired users?

Although Guest Services (previously known as WGS, or Wireless Guest Services) were designed for wireless (hotspot) users, Guest Services can also be employed for wired users on SonicOS Enhanced by placing the wired interface (or interfaces, as the case may be on the PRO 1260 with PortShield) into a Wireless Zone with "SonicPoint Enforcement" disabled. All Guest Services options then apply to wired users, including LHM, Dynamic Address Translations, Allow/Deny Networks, etc.

5. What is the difference between "authentication" and "authorization"?

"Authentication" describes the process of a user providing a response to some kind of challenge. The challenge can be just about anything, although traditionally it is a username:password. LHM breaks this dependence of the traditional model by abstracting the authentication. The role of authenticator is fulfilled by the LHM server, and the methods of authentication are bound only by imagination. Consider the following methods of authentication:

- Provide a valid username and password
- Guess the number the computer is thinking of
- Complete this questionnaire
- Pass a quiz with a score of at least 80%
- Click the "I Accept" button

Once authenticated, the client can then be authorized to do something. "Authorization" is the process of granting access to something. For authorization to be useful, the authorizer must have a means of stopping the client from getting to guarded resources. In the case of LHM, the SonicWALL is the client's gateway (either wired or wireless), so it can very effectively act as authorizer. Once the SonicWALL receives the OK from the authenticator for a client, it creates the Guest Services session and allows the client access to the Internet.

6. Can I use LHM to provide access using [LDAP, RADIUS, a button, the time of day, tasseography, a survey, relative barometric pressure, a passcode, etc.] as the authenticator?

Yes.

7. Can SonicWALL write the script for me that will do that?

We have provided a series of sample scripts as examples and for you to freely modify, but we do not provide custom scripts. We can, however, put you in touch with someone who can provide custom scripts. There are many SonicWALL partners who have web development teams on staff who can provide these services. More information will be provided once the program has been formalized.



Tech Note

8. I want to use the sample scripts SonicWALL provided. What do I need to do to use them?

Microsoft Windows 2000, XP, 2003 platform running IIS 5.0 or higher, running the latest service packs and hotfixes.

The Microsoft .NET 1.1 (or higher) Framework:

<http://www.microsoft.com/downloads/details.aspx?FamilyId=262D25E3-F589-4842-8157-034D1E7CF3A3&displaylang=en>

The latest .NET Framework Service Pack:

<http://www.microsoft.com/downloads/details.aspx?familyid=A8F5654F-088E-40B2-BBDB-A83353618B38&displaylang=en>

Copy the LHM script (or scripts) you wish to use to the wwwroot directory (usually in C:\inetpub\wwwroot).

Configure Guest Services on your SonicWALL to use External Guest Authentication, as described in the “[What do all the LHM settings mean?](#)” question that follows.

Some scripts need write privileges, particularly those that use databases. Depending on your configuration, two or three separate “users” will need to have write access to the script directories that require writing.

- The first account (all platforms) is **IUSR_MACHINENAME** (where machinename = the name of the local machine).
- The second account (on Windows XP) is **ASPNET** (ASP.NET machine account).
- The second account (on other platforms) is **IWAM_MACHINENAME** (where machinename = the name of the local machine).
- If database read/write access continues to fail even after assigning these permissions, it might be necessary to add read/write privileges for the **NETWORK SERVICE** account.

Note: Versions on .NET Framework prior to 1.1 had user permission problems on domain controllers (<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q315158>). It is strongly recommended that 1.1 (or higher) be installed.

After your environment is set up as described above, you will need to customize the scripts. We've tried to make this as simple as possible by placing all the interesting configurable bits in the “**myvars.aspx**” file. All entries are well commented, and their purposes and syntax should be evident. Further customization to the scripts themselves can be performed, but is generally not necessary.

9. Where can the LHM server reside?

The LHM Server can be virtually anywhere in the network, as long as it is reachable by the Guest Clients. It can be located at a centralized network operations center where it can administer LHM for multiple hotspots, or it can be co-located with a single SonicWALL security appliance.

10. Why are my Guest Clients unable to reach the LHM Server?

Guest clients communicate directly with the LHM server; the communication is not proxied by the SonicWALL security appliance. In other words:

- The Guest Client's subnet must be able to reach the LHM server.
- The LHM server must know how to reach the Guest Client's subnet (either by route, by NAT, or by VPN).
- Firewall Access Rules must be configured to allow the Guest Client subnet to reach the LHM server.

11. Why are the pages on the LHM server not loading?

As stated above, LHM requires the following connectivity between Guest Clients and the LHM server:

- The Guest Client's subnet must be able to reach the LHM server.
- The LHM server must know how to reach the Guest Client's subnet (either by route, by NAT, or by VPN).
- Firewall Access Rules must be configured to allow the Guest Client subnet to reach the LHM server.



Tech Note

12. How does the LHM exchange between the SonicWALL and the LHM server work?

- a) The Guest Client associates, gets a DHCP lease, and launches a web browser.
- b) DNS is allowed through the SonicWALL security appliance. The URL FQDN resolves to its IP address.
- c) The SonicWALL security appliance checks if the Guest Client has an authenticated session.
 - If it's new, it redirects the client to the internal redirect ("Please wait while you are being redirected") page.
- d) The internal redirect page attempts to redirect the Guest Client to the LHM server.
 - If it fails, it redirects the client to the internal server-down ("Wireless internet access is temporarily unavailable. Please click here to try again.") page.
- e) The Guest Client is redirected to the LHM server. In the redirect URL the SonicWALL embeds querystring information describing the embryonic session (e.g. the sessionID, the client's MAC and IP address, the SonicWALL's LHM management IP and port, the UFI, the originally requested URL).
 - The LHM server script grabs the querystring information.
 - The client directly retrieves the LHM landing page from the LHM server.
- f) Depending on the authorization model used (username:password, passcode, "I Accept" button, etc.) the LHM server decides that the Guest Client is worthy of access.
- g) The LHM server initiates a web-request to the SonicWALL security appliance at the configured management port (e.g. TCP 4043) to the "externalGuestLogin.cgi" page.
 - The LHM server POSTs the sessionID (which it grabbed in step e), along with the username (which it either got from the user or made up), and the session-lifetime and idle-timeout (both of which it determines).
- h) The SonicWALL security appliance validates the sessionID, and tries to create the session. It then responds to the POST with a result code describing whether or not it was able to authorize (create) the Guest session.
- i) The LHM Server interprets the result code, and reports the results (e.g. "Session Authorized – You may now start browsing.", "Session creation failed – Rats.", "Max sessions", etc.) to the Guest Client.

What do all the LHM settings mean? How do I configure them?

Rather than going into the full detail provided by the LHM specification document

(<http://www.sonicwall.com/support/pdfs/technotes/LHM.pdf>), let's just explain what the settings mean and how you might configure them:

The LHM configuration on SonicOS Standard and Enhanced is nearly identical. On SonicOS Standard, the "External Guest Authentication" page is reached from the "WGS > Settings" configuration pages, while on SonicOS Enhanced, it is reached from the [wireless] Zone configuration page, "Guest Services" tab. The settings are as follows:

General Tab: Local Web Server Settings

- Client Redirect Protocol – The protocol (HTTP or HTTPS) used by the SonicWALL security appliance when performing the initial internal client redirect via the "Please wait while you are being redirected" page. (This message configurable from the "Redirect Message" area on the "Web Content" tab.) This step is prior to redirection to the LHM server.

General Tab: External Web Server Settings

- Web Server Protocol – The protocol (HTTP or HTTPS) running on the LHM server.
- Web Server Host – The IP or resolvable FQDN of the LHM server.
- Web Server Port – The TCP port of operations for the selected protocol on the LHM server.
- Connection Timeout – The duration of time, in seconds, before the LHM server is considered unavailable on a redirect attempt. On timeout the client will be presented with the "Server Down" message configured on the "Web Content" tab.



Tech Note

General Tab: Message Authentication

- Enable Message Authentication – Use HMAC digest and embedded querystring in communication with the LHM server. Useful if you are concerned about message tampering when HTTP is used to communicate with the LHM server. Optional.
- Authentication Method – Select MD5 or SHA1.
- Shared Secret – The shared secret for the hashed MAC. If used, also needs to be configured on the LHM server scripts.

Auth Pages Tab: External Authentication Pages

- **Note:** These pages may each be a unique page on the LHM server, or they may all be the same page with a separate event handler for each status message. Examples will be provided below to work with the newly developed scripts.
- Login Page – The first page to which the client is redirected (e.g. “lhm/accept/default.aspx”).
- Session Expiration Page – The page to which the client is redirected when the session expires (e.g. “lhm/accept/default.aspx?cc=2”). After a session expires, the user must create a new LHM session.
- Idle Timeout Page – The page to which the client is redirected when the idle timer is exceeded (e.g. “lhm/accept/default.aspx?cc=3”). After the idle timer is exceeded, the user can log in again with the same credentials as long as there is time left of the session.
- Max Session Page – The page to which the client is redirected when the maximum number of sessions has been reached (e.g. “lhm/accept/default.aspx?cc=4”).

Web Content Tab: Redirect Message

- The default or customized message that will be presented to the client (usually for no more than one second) explaining that the session is being redirected to the LHM server. This interstitial page is used (rather than going directly to the LHM server) so that the SonicWALL security appliance can verify the availability of the LHM server.

Web Content Tab: Server Down Message

- The default or customized message that will be presented to the client if the Redirector determines that the LHM server is unavailable.

Advanced Tab: Auto Session Logout (optional)

- The time increment and the page to which the SonicWALL security appliance will POST when a session is logged out (either automatically or manually).

Advanced Tab: Server Status Check (optional)

- The time increment and the page to which the SonicWALL will POST to determine the availability of components on or behind (e.g. a back-end database) the LHM server.

Advanced Tab: Session Synchronization (optional)

- The time increment and the page to which the SonicWALL will POST the entire Guest Services session table. This allows the LHM server to synchronize the state of Guest Users for the purposes of accounting, billing, or mere curiosity.

13. Can I change the LHM Management port from its default of TCP 4043?

Yes. This is easily done on SonicOS Standard from the LHM configuration pages under WGS. On SonicOS Enhanced, it can be changed by modifying the port values of the “External Guest Authentication” Service Object.



Tech Note

14. Do I need to use the HMAC option? If I do want to use it, how do I use it?

The HMAC function is optional. Its purpose is to ensure that messages sent by the SonicWALL to the LHM server, and the LHM server to the SonicWALL security appliance have not been tampered with. It achieves this by calculating a keyed (password-aided) message authentication code on the information being passed between the two peers, and by adding that calculated digest to the data. When the other side receives the data, it calculates the digest itself, and compares it to the transmitted MAC; if the two match, that proves the data was delivered intact. You should consider using the HMAC option if you are in an insecure environment, if you are overly concerned with security, or if you have been diagnosed with escalated dopaminergic activity.

If you choose to use the HMAC, you may implement your own HMAC routines, but the simplest method is to use the SonicWALL-provided SonicSSL.dll library, along with the libeay32.dll - the latter is freely available as part of OpenSSL, the former was written by SonicWALL, and both are available from SonicWALL by request.

Copy the libeay32.dll file to the path on the LHM (IIS) server (for example, into the C:\Windows\system32 folder), and copy the SonicSSL.dll file to any location on the same server. Register the SonicSSL.dll file with the command "**regsvr32 SonicSSL.dll**". Once this is done, the LHM scripts will be able to use the `Server.CreateObject("SonicSSL.Crypto")` object for HMAC calculations. The HMAC functions are included in the scripts described in this document.

One important note on the HMAC function is that the SonicWALL security appliance URL Encodes (converts certain characters from their ASCII notation to hex notation) the "req" (originally requested URL) portion of the querystring, but the SonicWALL method of URL encoding is slightly different from the Microsoft method (as employed by `Request.QueryString`, for example). Because of this difference in methods, it is possible for the string upon which the HMAC is being performed to be different between the SonicWALL and the LHM server. The provided scripts compensate for this by manually encoding the "req" portion of the querystring in a fashion consistent with the SonicWALL method.

15. Does SonicWALL provide any support for these scripts?

The scripts are provided as examples, and they are not supported by SonicWALL Technical Support, nor can SonicWALL support assist with the configuration of your LHM back-end environment. Future consultative support services might address this.

16. I've written a new script, I've made some great enhancements to your scripts, or I've just made your scripts work a whole lot better than you did – is SonicWALL interested?

Yes! We are always looking for new ways to use LHM, and for people to contribute to the library of available scripts. We will consider LHM scripts written on any platform, using any authentication method. Please send an email to products@sonicwall.com describing your script, and we will consider it for addition to our library. Submitting a script gives SonicWALL permission to freely modify and/or redistribute the submitted script.



Tech Note

LHM Script Library

The SonicWALL LHM Script library was established to serve as a resource for people using or wishing to use LHM for Guest Services. The goal is to attract multiple contributors and consumers, helping the library to grow to house a large, varied, and useful collection of scripts that anyone can modify or use as-is.

The first contribution to the library comprises six scripts – three in response to common user requests ('accept', 'guestbook', and 'adauth'), and three that just seemed fun ('lhmquiz', 'random', and 'paypal'). They were written outside of a Visual Studio .NET development environment, so the style can be classified as functionally insane.

Common to all the scripts is modularization of the configurable variables, such as the paths to files, server IP addresses, use of a popup logout window, salt values, and timer settings. These configurable values are gathered into the **"myvars.aspx"** file, so that per-environment editing can be done in one place rather than having to search for configurable elements. Also common to all the scripts is extensive commentary explaining step-by-step what is being done.

A **"chooser.aspx"** landing page has been provided at the top-level of the scripts directory. This script was designed for demonstration environments to allow for the selection of a lower-level (specific) script without having to reconfigure the LHM settings on the SonicWALL to point to a specific script. In other words, LHM on the SonicWALL can be configured to point to the top-level chooser.aspx script, which will then enumerate all the sub-directories (lower-level scripts: 'random', 'accept', 'adauth', etc.). The top-level chooser.aspx script will open the target lower-level default.aspx script in a new window, and will pass the original querystring in its entirety.

All of the scripts begin with the **"default.aspx"** page, and client redirection is performed automatically as needed. The LHM configuration on the SonicWALL should, therefore, point to the default.aspx page at the appropriate path (e.g. "lhm/accept/default.aspx" or "lhm/adauth/default.aspx"). Some scripts will have separate administrative function pages - these will be noted in the script descriptions.



Tech Note

A “**logout.aspx**” page is also provided with each script. The use of this page is controllable with the “logoutPopup” variable in myvars. Setting a value of “1” will enable the use of the popup logout window. The window is invoked by the LHM authentication process after a successful response code (50) is received from the SonicWALL. The script passes the sessID, mgmtBaseUrl, and sessTimer variables to the logout.aspx window so that the window can track the session time, and can POST a logout event back to the SonicWALL (at the mgmtBaseUrl) for the correct session (sessID) when/if the user wants to manually terminate the session. A few notes about the use of the logout popup window:

1. The use of the logout popup is not necessary. Sessions will timeout by themselves after their configured lifetime expires. The popup window simply provides fastidious users a mechanism to manually terminate their own sessions.
 2. The window launches with a javascript popup, so popup blockers will block the window.
 3. Closing the window will not interrupt the session. Only the “Logout” button can end a session.
 4. Since the countdown timer runs client-side, steps have been taken to prevent refreshing the page. Refreshing the page will reset the client-side countdown timer, but it will not affect the actual session timer. The F5 key and right-click mouse event are captured and suppressed – this does not work on all browsers.
5. The use of the logout popup should agree with the nature of the scripts authentication scheme:
- a. Some scripts have non-exclusive login processes, meaning that the user can login repeatedly (such as the ‘Accept’ and ‘ADAuth’ scripts). The use of the logout popup on these non-exclusive scripts is encouraged.
 - b. Some scripts are non-exclusive, but gather data that should be kept unique (such as the ‘Guestbook’ and ‘LHMQuiz’ scripts). The use of the logout popup on these scripts is acceptable, but can lead to redundant data being gathered.
 - c. Some scripts are exclusive, meaning that once the user authenticates, it will not be possible to repeat the authentication process without some kind of cost (such as the ‘PayPal’ script, or the ‘Random’ script where useDB is enabled). The use of the logout popup is discouraged on these scripts since the user will have no simple means of logging back in.

The scripts also provide hidden output in the event of a .NET procedure error, where the text is hidden by matching it to the color of the background. In the event of some kind of failure or error condition, error output may be provided and made visible by hitting CTRL-A on the web-page to select all of the text.

The following is a description of each of the scripts, what they do, and how they do it. Some of the descriptions below will highlight certain myvars variables in green; those that are highlighted *must* be configured with values appropriate to your environment or the scripts will fail to run properly.

As new scripts get added to the library, similar descriptions will accompany them to help with understanding, customization, and integration.

The LHM Script Library describes the following scripts:

[Accept Script](#)
[ADAuth Script](#)
[Guestbook Script](#)
[LHMQuiz Script](#)
[PayPal Script](#)
[Random Script](#)



Tech Note

Accept Script

| | | |
|----------------------------------|--|---|
| Authentication Model | The Guest Client clicks the "I Accept" button. | |
| Purpose | Present an acceptable use policy, terms of service, or welcome screen to the client. | |
| myvars Variables | logoutPopup | Controls the use of the logout popup window. Set to 0 to disable the popup window, set to 1 to enable the popup window. |
| | sessTimer | The session timer in seconds. |
| | idleTimer | The idle timer in seconds. |
| | username | The username applied to the guest sessions. Since the script does not grab a username from the client, it can be explicitly set here for all clients, or it can be set to useMAC to set the username to the MAC address. |
| | strHmac | The shared secret for the optional HMAC function. |
| | hmacType | The digest type to use if HMAC is in use, either MD5 or SHA1 . |
| | logo | The name of the logo (image) file to use on page headers. |
| Session Flow | <ol style="list-style-type: none">1. The Guest Client clicks the I Accept button.2. The LHM post string is assembled with the sessionId, the username (either default of MAC), the default session lifetime, and idle lifetime.3. The script performs the LHM post to the SonicWALL to authorize the session. | |
| Additional Considerations | Only the basic LHM configuration is required. | |

Tech Note

ADAuth Script

| | | |
|----------------------------------|--|--|
| Authentication Model | The Guest Client provides their username and password. These credentials are then authenticated against an Active Directory or LDAP database. | |
| Purpose | Classical authorization model using Active Directory via LDAP. Support for per-user session-timer and idle-timer setting provided by optionally grabbing LDAP attributes from the database during authorization. | |
| myvars Variables | logoutPopup | Controls the use of the logout popup window. Set to 0 to disable the popup window, set to 1 to enable the popup window. |
| | myLdapServer | The IP address or resolvable FQDN of the LDAP/AD server providing authentication. |
| | myLdapDomain | The LDAP/AD domain name |
| | retrAttr | Specifies whether or not to retrieve session and idle timer values from the authenticating user's LDAP attributes (attributes defined later). Set to 0 to disable retrieval; set to 1 to attempt retrieval. |
| | useCN | If reAttr=1, then this flag sets whether to use the common name (cn) to retrieve attributes, or the AD default login name (sAMAccountName). Set to 1 to use cn. When authenticating against AD, this flag should be set to 0 . |
| | sessAttr | The LDAP attribute from which to retrieve the session timer (in seconds). If no value can be retrieved, or if the retrieved value is not numeric, the default session timer (defined below) will be used. |
| | idleAttr | The LDAP attribute from which to retrieve the idle timer (in seconds). If no value can be retrieved, or if the retrieved value is not numeric, the default idle timer (defined below) will be used. |
| | sessTimer | The default session timer in seconds. |
| | idleTimer | The default idle timer in seconds. |
| | strHmac | The shared secret for the optional HMAC function. |
| | hmacType | The digest type to use if HMAC is in use, either MD5 or SHA1 . |
| | logo | The name of the logo (image) file to use on page headers. |
| Session Flow | <ol style="list-style-type: none"> 1. The Guest Client enters their LDAP/AD username and password. 2. The provided credentials are used to bind with the configured LDAP server. 3. If the bind attempt succeeds, the user is authenticated. 4. If the reAttr flag is set, an attempt is made to retrieve the defined sessAttr and idleAttr attributes (e.g. "pager" and "mobile") from the LDAP DB. If valid results are retrieved, they will be used, otherwise the default values will be used. 5. The script performs the LHM post to the SonicWALL to authorize the session. | |
| Additional Considerations | Requires that the LHM server be able to communicate with the configured LDAP/AD server, either by route, NAT, or VPN. If the reAttr option is used, it requires that the LDAP attributes be defined for user-specific values to take effect. (Note: the 'pager' and 'mobile' attributes were selected because they are not frequently used, and because they can be set directly through Microsoft's Users and Computers MMC.) | |

Tech Note

Guestbook Script

| | | |
|----------------------------------|---|---|
| Authentication Model | The Guest Client provides their name, address, phone, email, URL (optional), and comment (optional) information. | |
| Purpose | Gather market information, write the information to a database for later use. | |
| myvars Variables | logoutPopup | Controls the use of the logout popup window. Set to 0 to disable the popup window, set to 1 to enable the popup window. |
| | sessTimer | The session timer in seconds. |
| | idleTimer | The idle timer in seconds. |
| | strHmac | The shared secret for the optional HMAC function. |
| | hmacType | The digest type to use if HMAC is in use, either MD5 or SHA1 . |
| | logo | The name of the logo (image) file to use on page headers. |
| Session Flow | <ol style="list-style-type: none">1. The Guest Client enters their personal information and clicks Submit.2. The entered information is written to a local .mdb database file for later use.3. The LHM post string is assembled with the sessionID, the username (as provided in the web-form), the default session lifetime and idle lifetime.4. The script performs the LHM post to the SonicWALL to authorize the session. | |
| Additional Considerations | Because the script will be writing to the database, it will be necessary to configure write privileges for the IUSR_MACHINENAME and IWAM_MACHINENAME (or ASPNET) accounts, as described in FAQ question #8, I want to use the sample scripts SonicWALL provided. What do I need to do to use them? | |

LHMQuiz Script

| | | |
|-----------------------------|--|--|
| Authentication Model | The Guest Client takes a quiz. A passing score serves as the authentication credentials | |
| Purpose | It is common for network access to be provided in a classroom environment. By using a passing score on a test of the material being taught as the method for authentication, an instructor can ensure that the course material has been mastered before the irresistible temptation of the Internet diverts attention. The script also emails the completed passing test to the test-taker, and mails failing tests to the proctor/instructor. | |
| Myvars Variables | logoutPopup | Controls the use of the logout popup window. Set to 0 to disable the popup window, set to 1 to enable the popup window. |
| | passingScore | The score (an integer representing a percentage) required to pass the quiz. |
| | quizFile | The filename for the XML source for the quiz (e.g. quiz.xml, shortquiz.xml). |
| | quizName | The name of the quiz, used throughout the script. |
| | quizFrom | The From: email address that will be used when emailing the quiz. |
| | quizTo | The To: email address that failing quizzes are to be sent to (e.g. the test proctor or instructor). |
| | imagePath | The email will include an attachment for the correct and incorrect answers. This sets the path for those image files. This is generally set to the same path of the script files themselves. |
| | smtpServer | The IP address or resolvable FQDN of the SMTP server to be used for quiz result delivery. This can be set to 127.0.0.1 if the local IIS SMTP server instance it to be used. |
| | sessTimer | The session timer in seconds. |
| | idleTimer | The idle timer in seconds. |
| | strHmac | The shared secret for the optional HMAC function. |
| | hmacType | The digest type to use if HMAC is in use, either MD5 or SHA1 . |



Tech Note

| | | |
|----------------------------------|--|---|
| | logo | The name of the logo (image) file to use on page headers. |
| Session Flow | <ol style="list-style-type: none"> 1) The Guest Client is prompted to enter their full name and email address. A correct/valid email address is required for delivery of the completed passing quiz. 2) After entering name and email, the Guest Client is redirected to the quiz.aspx page. This is where the multiple choice test is administered. 3) The test questions themselves are contained in the quiz.xml file, defined by the quiz.xsd (XML Schema Definition) file. The quiz.xml file can and should be edited to customize the quiz, but the quiz.xsd document should not be edited unless absolutely necessary. <ol style="list-style-type: none"> a) Two versions of the quiz are included: quiz.xml (containing 10 questions) and shortquiz.xml (containing 2 questions, for testing that the script works). The quiz will support any number of questions, and each question will support any number of answers, one of which must be marked the 'correct' answer, with <i>correct="yes"</i>. It should be fairly straightforward to modify the provided quiz.xml file as needed. 4) At the end of the quiz, the results are shown. <ol style="list-style-type: none"> a) If it is a failing score, the test results are emailed to the instructor (email address defined in myvars), and the Guest Client is prompted to take the test again. The LHM session will not be authorized. b) If it is a passing score, the test results are emailed to the test-taker, and the LHM session will be authorized. c) The emailed test is sent in an HTML format, and includes the "checkmark.gif" and "block.gif" (right and wrong) graphics as attachment so that they can be displayed in the email. 5) If the test was passed, the LHM post string is assembled with the sessionId, the username (as provided in the web-form), the default session lifetime and idle lifetime. 6) The script performs the LHM post to the SonicWALL to authorize the session. | |
| Additional Considerations | <p>Access to an SMTP server is required to deliver the test results. Since the script will be relaying the mail through the server, the SMTP server will need to be configured to allow relaying from the LHM server. This is best accomplished by configuring the SMTP server to allow relaying from the IP address of the LHM server.</p> <p>Most IIS installations include a local SMTP server, so it is convenient to use this local SMTP server for mail delivery by configuring the "smtpServer" variable in myvars as 127.0.0.1. Even when using the local SMTP server for mail delivery, it is necessary to allow relaying. In most configurations, this is performed by going into the IIS MMC configurator, then right clicking on Default SMTP Virtual Server, selecting Properties, selecting the Access tab, clicking the Relay button, and adding 127.0.0.1 to the access granted list. When using a non-local SMTP server, that SMTP server should be configured to allow the LHM server to relay by its actual IP address.</p> | |

Tech Note

PayPal Script

| | |
|-----------------------------|---|
| Authentication Model | The Guest Client buys 1 hour or 24 hour access with a Buy Now button using their PayPal account. Payment is made through PayPal to the hotspot provider's PayPal merchant account. |
| Purpose | <p>Nearly everyone who buys or sells on the Internet uses PayPal. It is very easy to setup a "buyer" account, and to link it to any form of payment (credit-card, bank-card, checking account, etc.) and it is almost equally easy to "upgrade" a buyer-only account to a "merchant" account. Having a merchant account allows PayPal users to accept payment from other PayPal users for goods or services. The funds transfer is run through PayPal, providing merchants a way to do business online, accepting any form of payment, without having to setup any sort of complicated payment processing. This eliminates what is perhaps the single biggest obstacle to being a fee-based hotspot provider.</p> <p>Paypal provides a feature called the "Buy Now Button." This feature allows for one-click-ish transactions. The buttons are forms, generated with the assistance of PayPal, that contain information about the item or service being purchased. When the buyer clicks on the Buy Now Button, the session is redirected to the PayPal site with a querystring containing all the details of the transaction (the seller, the item, the price, etc.). Rather than using the basic Buy Now button (which is client-side rather than server-side code), the PayPal script uses a custom, server-side Buy Now routine.</p> <p>Also included in the Buy Now redirect is the path for the auto-return. Auto-return is a PayPal feature that sends the buyer back to the merchant's site after the PayPal transaction. Auto-return is required when using PDT (described below). The custom Buy Now redirect also embeds the LHM sessionId and the mgmtBaseUrl into a custom string in the Buy Now redirect to PayPal. This allows us to track the session even though it leaves the LHM server, goes to PayPal, and then comes back (via auto-return for PDT).</p> <p>The basic PayPal payment system provides notification of payment to merchants by email. This is acceptable for physical goods because the purchase/ship transaction does not have to occur in real-time; the merchant can wait hours or days for the notification before shipping the product. For transactions that require instantaneous delivery, such as buying hotspot access, a more real-time method of payment is required.</p> <p>PayPal offers two methods of payment notification. One is called Instant Payment Notification (IPN), which works by PayPal making a web-services call to the merchant's site indicating that payment for a particular transaction has cleared. Unfortunately, this does not always occur in real-time (it can take up to 20 minutes for this asynchronous notification to arrive) so it was not employed in this script. (More can be read about IPN at https://www.paypal.com/cgi-bin/webscr?cmd=p/xcl/rec/ipn-intro-outside)</p> <p>The other method is called Payment Data Transfer (PDT – see http://www.paypal.com/cgi-bin/webscr?cmd=p/xcl/rec/pdt-intro-outside). This method occurs in absolute real-time using PayPal's auto-return method. PDT provides instant notification to the merchant of the state of a transaction (either SUCCESS or FAIL), as well as of the payment_status (Completed, Pending, Denied, Failed, Refunded, Reversed, or Cancelled_Reversal). By instantly knowing the status of the transaction and the payment, it is possible to immediately provide service without the risk of losing payment.</p> |

Tech Note

| | | |
|-----------------------------------|----------------------------|---|
| myvars Variables | logoutPopup | Controls the use of the logout popup window. Set to 0 to disable the popup window, set to 1 to enable the popup window. |
| | debugFlag | Sets the debug output for the PayPal PDT transfer. 0 =Off, 1 =On. |
| | pdtPath | The path to which the Guest Client is redirected by the PDT auto-return (described above in the 'Purpose' section). |
| | paypalCGI | The URL for the PayPal CGI serving as the gateway for the PayPal transaction. The URL itself should not be changed, but there are two options: either the "live" (real) PayPal site, or the paypal sandbox (part of the PayPal developer network), which can be used for testing. |
| | myBusiness | The email address (how PayPal recognizes the "business") of the hotspot provider. This must match the email address of the merchant account that will be receiving payment for the transactions. |
| | token | The Payment Data Transfer option generates a unique token for each merchant. This is where you specify your PayPal-provided unique token. The token must be correct, or the PDT transaction (not the actual PayPal transaction) will fail. |
| | itemName1 itemName2 | The names of the two access options, e.g. "1 Hour Secure Internet Access" and "24 Hours Secure Internet Access". |
| | itemNumber1 itemNumber2 | The item number (a mostly arbitrary internal PayPal reference) for the two access options, e.g. "1hour" and "24hour". |
| | itemTimer1 itemTimer2 | The session timer in seconds for the two access options, e.g. "3600" for 1 hour and "86400" for 24 hours. |
| | itemAmount1 itemAmount2 | The price in US dollars for the two access options, e.g. "0.01" (one cent) and "0.02" (two cents). Limited time promotional bargain pricing. |
| | itemButton1 itemButton2 | The button text for the two access options, e.g. "1 Hour Access - \$0.01" and "24 Hours Access - \$0.02". |
| | strHmac | The shared secret for the optional HMAC function. |
| | hmacType | The digest type to use if HMAC is in use, either MD5 or SHA1 . |
| | logo | The name of the logo (image) file to use on page headers. |

Tech Note

| | |
|---------------------|---|
| Session Flow | <ol style="list-style-type: none"> 1) The Guest Client launches their web-browser, and is redirected by LHM to http://lhmserver/paypal/default.aspx 2) Guest client (buyer) clicks on one of the "Buy Now" buttons, e.g. "1 Hour Access - \$0.01." 3) The client is redirected to the PayPal site with a querystring containing all the information about the merchant, the item, the LHM session (stuffed into the custom variable), and the auto-return URL (defined in myvars as "pdtPath"). <ol style="list-style-type: none"> a) The pdtPath resides on the LHM server. The path should be the same as the default.aspx path (as configured on the SonicWALL security appliance), but should point to the pdt.aspx file. This way, when the PayPal transaction is completed and PayPal redirects the client back to the merchant site, the client will be redirected back to the http://lhmserver/paypal/pdt.aspx page. (HTTP can be used on the LHM Server since no sensitive information is entered on the LHM server itself – the PayPal transaction occurs via HTTPS directly between the Guest Client and PayPal). <p>A sample Buy Now redirect string:</p> <p>https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_xclick&business=demo@sonicwall.com&item_name=1%20Hour%20Access&item_number=1hour&amount=0.01&currency_code=USD&lc=US&bn=PP-BuyNowBF&no_note=1&no_shipping=1&cancel_return=http://lhmserverpaypal/default.aspx&return=http://lhmserver/lhm/paypal/pdt.aspx&custom=35378e67833faa3de83aa3b771https%3a%2f%2f172.16.17.1%3a4043%2f</p> 4) The Guest Client logs into PayPal (or creates a new account, as needed) and completes the transaction with PayPal. Once the transaction is completed, the client is redirected back to http://lhmserver/pdt.aspx. Included in the redirect is a querystring containing the transaction id (tx), the status (st), the amount (amt), the currency type (cc), the custom value (cm), and an encrypted signature (sig). A sample redirect string: http://lhmserver/lhm/paypal/pdt.aspx?tx=4LN76482JF4605045&st=Completed&amt=0.01&cc=USD&cm=35378e67833faa3b771https%3a%2f%2f172.16.17.1%3a4043%2f&sig=qdsNC4fIKwtPvigoGAXCpeV9gS%2f2E%2bGGVbTZ3STrUV1Ci9K3c2zTdJMuuKcmRiif1SybsZtUqDYqzzfMg64AF3PKCk85rrPubYT4K4aC 5) The Guest Client accessing the pdt.aspx script at the URL above starts the PDT process on the LHM server. The script builds a querystring consisting on "cmd=_notify-synch" (indicating that it is a PDT transaction) along with the "tx" (transaction ID) and the "at" variable set to the merchant's token (defined in myvars). This is then POSTed to the "paypalCGI" URL (as defined in myvars). 6) PayPal responds to the POST with a SUCCESS or a FAIL code. <ol style="list-style-type: none"> a) If it is FAIL, the script indicates to the client that the PayPal transaction fails, and they are prompted to seek assistance. b) If it is SUCCESS, it also provides detail about the transaction, as follows: <pre> SUCCESS txn_type=web_accept payment_date=00%3A39%3A48+Oct+30%2C+2005+PDT last_name=Niquai item_name=1+Hour+Secure+Internet+Access payment_gross=0.01 mc_currency=USD business=lhmdemo%40sonicwall.com payment_type=instant payer_status=verified tax=0.00 payer_email=lhmClient%40sonicwall.com txn_id=84K306380G150640T quantity=1 receiver_email=lhmdemo%40sonicwall.com first_name=Sah payer_id=XWRZGABD6UV2W receiver_id=REW4W5WANU294 item_number=1hour payment status=Completed </pre> |
|---------------------|---|



Tech Note

| | |
|----------------------------------|--|
| Additional Considerations | <p>Requires a PayPal merchant account.</p> <p>Requires that the PayPal account be setup for auto-return and for PDT (see http://www.paypal.com/cgi-bin/webscr?cmd=p/xcl/rec/pdt-techview-outside)</p> <p>For testing purposes, it is strongly suggested that a (free) PayPal sandbox account be setup through the PayPal Developer's Network (https://developer.paypal.com) and (https://www.sandbox.paypal.com)</p> <p>Important: Since the Guest Client will be redirected directly to the PayPal site, ALL PayPal site IP addresses must be setup on the SonicWALL as Allowed Networks on the Guest Services configuration. These include the following:</p> <p>www.paypal.com</p> <ul style="list-style-type: none">64.4.241.3264.4.241.33216.113.188.32216.113.188.35216.113.188.66216.113.188.67 <p>www.paypalobjects.com</p> <ul style="list-style-type: none">216.113.188.2564.4.241.62216.113.188.9 <p>www.sandbox.paypal.com</p> <ul style="list-style-type: none">66.135.197.160 <p>developer.paypal.com</p> <ul style="list-style-type: none">66.135.197.163 <p>(Really can't wait for those FQDN Address Objects...)</p> |
|----------------------------------|--|

Tech Note

Random Script

| | |
|-----------------------------|--|
| Authentication Model | The Guest Client enters an algorithmically validated, randomly generated passcode. |
|-----------------------------|--|

Tech Note

| | |
|----------------|---|
| Purpose | <p>Traditional passcode authentication requires that a passcode be generated prior to use and stored on the authenticating platform. For example, Wireless Guest Services requires that accounts be generated on the particular SonicWALL security appliance on which they will be used. The Random script eliminates this dependency by using a salted algorithm to generate and validate passcodes. This means that passcodes never have to be stored anywhere, and as long as the salt is the same, passcodes are completely migratory (that is, they can be used at any site, even against different LHM servers).</p> <p>The practical implication of this is that guest account passcodes can be generated in bulk, distributed, and used at any time in the future. For example, passcodes could be generated (using a particular salt), printed (on to certificates, business cards, scratch cards, etc.) distributed and used at any site whose LHM server employs the same algorithmic salt. The passcodes could be given an absolute (rather than relative) expiration date, at which time the salt can be changed to invalidate the expired passcodes.</p> <p>The same way that a common salt can be used to validate a set of passcodes across multiple site, unique salts can ensure that passcodes generated at one site cannot be used at another with a dissimilar salt; so although a common algorithm is used to generate and validate all passcodes, the addition of the salt to the hash function provides uniqueness as needed.</p> <p>In addition to the default.aspx script is a generator.aspx script. The generator script is where the passcodes are generated. Anywhere from 1 to 999 passcodes may be generated at once. After generation, individual passcodes can be printed, or the entire list can be exported to a .csv file.</p> <p>Support was included for two classes of passcodes: 1 hour and 24 hour. Either type of passcode can be generated by the generator script.</p> <p>The generation algorithm works as follows:</p> <ol style="list-style-type: none">1) Generate a random code (root-passcode) of "randChars" (integer with a default value of 6) characters, as defined in myvars. The character set for the random code generator can be modified within the default.aspx file.2) The salt (defined in myvars as the "salt" string) is prefixed to the root-passcode.3) A SHA1 hash is then calculated on the resulting string. Three pairs of characters are then grabbed from the hash:<ol style="list-style-type: none">a) For a 1-hour passcode, the 408 pair are grabbed (characters 4,5 + 0,1 + 8,9).b) For a 24-hour passcode, the 752 pair are grabbed (characters 7,8 + 5,6 + 2,3).4) The 6 characters chosen from the hash are then concatenated to root-passcode.5) The result is the distributable passcode. <p>The validation algorithm works in reverse:</p> <ol style="list-style-type: none">1) Guest client enters their passcode (call this enteredCode).2) The script grabs the first "randChars" characters of the entered code (call this root-passcode).3) The salt is prefixed to the root-passcode, and a SHA1 hash is calculated:<ol style="list-style-type: none">a) The 408 pair of characters are grabbed and attached to the root-passcode. If this matches the enteredCode, then it is validated as a 1-hour passcode.b) If the 408 pair did not match, then the 752 pair is tried. If this matches the enteredCode, then it is validated as a 24-hour passcode.c) If neither match, then the code is not valid. <p>Once the enteredCode has been validated, the "usedcodes.mdb" database is queried to see if the code has already been used. If the enteredCode is not found in the database, the LHM session authorization sequence commences, using the MAC address as the userName. After the LHM session is authorized and an acknowledgement has been received by the LHM server, the root-passcode from the enteredCode is written to the usedcodes.mdb database so that it cannot be re-used. When (if) the salt is changed, it would be advisable to flush the database.</p> |
|----------------|---|

Tech Note

| | | |
|----------------------------------|--|--|
| myvars Variables | logoutPopup | Controls the use of the logout popup window. Set to 0 to disable the popup window, set to 1 to enable the popup window. |
| | useDB | Controls the use of the used passcode database. If useDB = 0 , then the database will not be read from or written to, allowing passcodes to be used repeatedly. If useDB = 1 , then used passcodes will be written to the database, and new authentication processes will check the database to determine if passcodes have already been used. |
| | randChars | The number of random characters to include in the root-passcode. The default is 6. This will result in 12 character passcodes since the hash component always adds an additional 6 characters. |
| | salt | The salt to use in computing the hash. Be sure to use a good salt to prevent unwanted passcode migration/collisions. |
| | sessTimer | The session timer in seconds. |
| | idleTimer | The idle timer in seconds. |
| | strHmac | The shared secret for the optional HMAC function. |
| | hmacType | The digest type to use if HMAC is in use, either MD5 or SHA1 . |
| | logo | The name of the logo (image) file to use on page headers. |
| Session Flow | <ol style="list-style-type: none"> 1. The Guest Client enters their passcode. 2. The passcode is validated using algorithmic validation, described in the "Purpose" section above. 3. If the code is validated, it is checked for previous use in the usedcodes.mdb database. 4. If it is not present, the LHM session (either 1-hour or 24-hours) is initiated, using the MAC address as the username. 5. After the LHM session is initiated, the script writes the root-passcode to the usedcodes.mdb database so that it cannot be reused. 6. The script performs the LHM post to the SonicWALL to authorize the session. | |
| Additional Considerations | <p>Since the script will be writing to the database, it will be necessary to configure write privileges for the IUSR_MACHINENAME and IWAM_MACHINENAME (or ASPNET) accounts, as described in FAQ question #8, I want to use the sample scripts SonicWALL provided. What do I need to do to use them?</p> <p>The generator.aspx script should be located in a secure (publicly inaccessible) area on the web-server.</p> | |